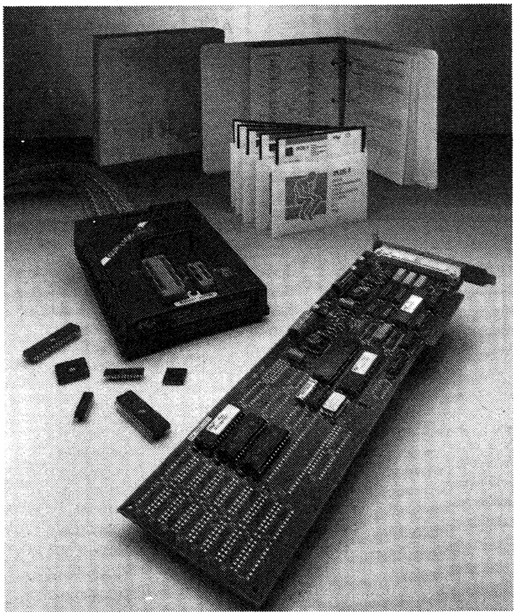
iPLDS II
THE INTEL PROGRAMMABLE LOGIC DEVELOPMENT SYSTEM VERSION II

- **Hardware and Software Necessary to Turn Design Concepts into Functional Erasable Programmable Logic Devices (EPLDs)**
- **Menu-driven Software with On-line Help Messages for All Stages of the Design Process**
- **iUP-PC Hardware Programs Intel EPLD's, EPROM's, E^2PROM's, Peripherals, and Microcontrollers with one PC-based System**
- **All Equipment Interfaces with the IBM PC/XT*, PC/AT*, and True Compatibles**
- **JEDEC Standard Design File, Part Utilization Report, Minimized Equation File, and Compiler Error File All Available as Outputs**
- **Supports a Variety of Input Methods:**
  Schematic Entry
  — TTL Library
  — EPLD Primitives Library
  Text Editor Entry
  — State Machine
  — Boolean Equations
- **Macro Expander Accepts TTL, and User-Defined Macros and Expands Them into Equivalent EPLD Primitives**
- **Espresso** Minimizer Reduces Logic Equations to Least Number of Product Terms**
- **Supports All Intel EPLD's Including the 5AC312, 5AC324, and 85C508**

Release 2.0 of Intel's Programmable Logic Development System II (iPLDS II) is a powerful set of tools for transforming a logic design into customized silicon. The system provides design entry, logic compilation, and device programming capability on a desktop using an IBM PC/XT, PC/AT, or compatible.



290134–1

iPLDS II Components Picture

*IBM PC/XT, PC/AT are registered trademarks of International Business Machines Corporation.

**ESPRESSO is a copyrighted by the University of California at Berkeley and is used with permission.

October 1988
Order Number: 290134-004

INTRODUCTION TO PROGRAMMABLE LOGIC DESIGN

When performing a programmable logic design on a CAD system, the design must first be entered using one of a variety of entry methods. These methods typically include schematic capture or Boolean equation entry using a standard text editor. Less typical entry methods include netlist entry, whereby a hand drawn schematic can be entered in a node-by-node fashion, or state machine entry in a text or graphical mode.

Once the design has been entered into the CAD package, several processing steps may occur. The design is usually translated into a format usable by the software, logic reduction may be performed, and, finally, some form of programming file can be produced. Most CAD packages also produce documentation of the minimization and device fitting results, including the final pin assignments.

Once the programming file has been generated, the design can be transferred into silicon in a programming manner similar to that used for EPROMs.

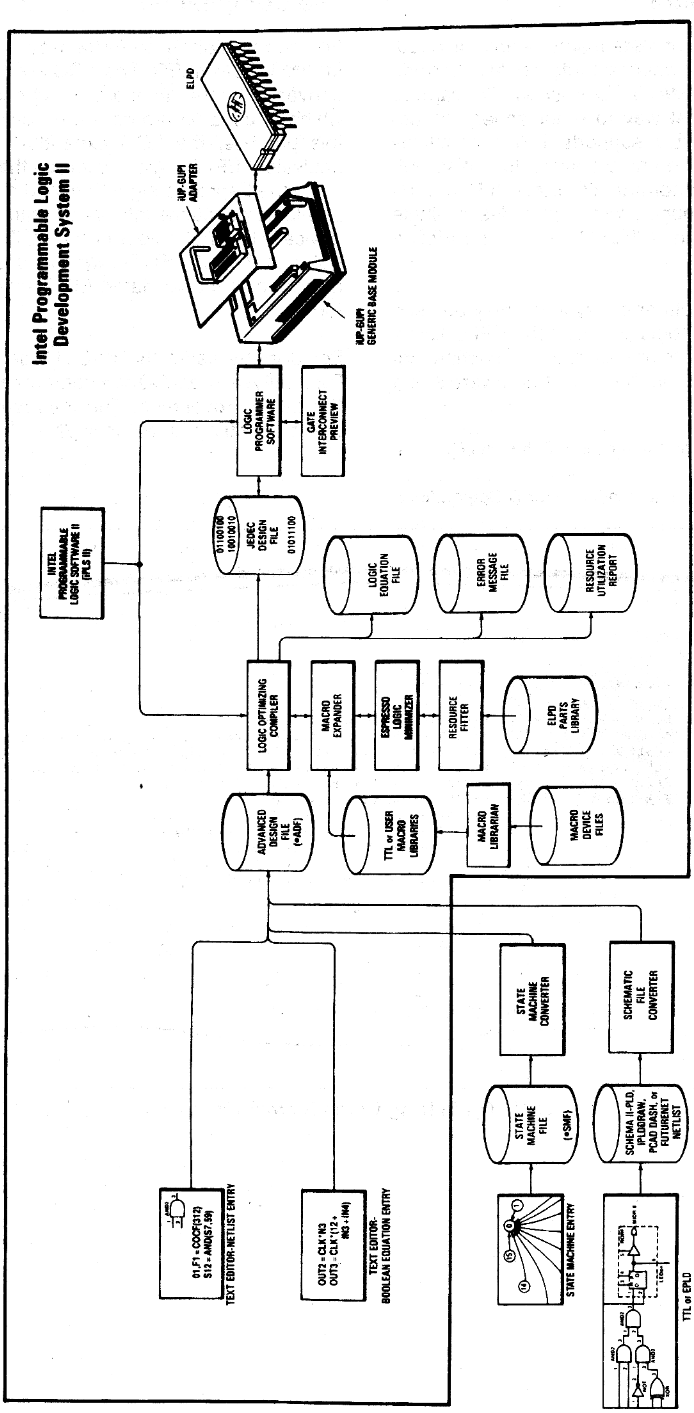
FUNCTIONAL DESCRIPTION OF iPLDS II

All of the design entry methods with the exception of graphic state machine entry are supported by the iPLDS II software. iPLDS II supports netlist and Boolean equation entry using any standard text editor. State machine software and schematic capture libraries are also available from Intel as optional entry methods. Depending on the entry format used, the design may require translation into Advanced Design File (ADF) format. Once the design is in ADF form, the Logic Optimizing Compiler expands any macros, minimizes all equations, and fits the design into a device-specific JEDEC Design File. The JEDEC Design File is programmed into the EPLD by the Logic Programmer Software using the iUP-PC hardware. Thus, the circuit design is transformed into an operating EPLD on one workstation.

The Intel Programmable Logic Software II (iPLS II) is composed of four functional modules: design entry, netlist conversion, file compilation and device programming.

Design Entry

Design entry is typically accomplished by creating an ADF using an ASCII text editor, or by using a schematic capture package.


Intel Programmable Logic Development System II
ELPD
iUP-GUPI ADAPTER
iUP-GUPI GENERIC BASE MODULE
LOGIC PROGRAMMER SOFTWARE
GATE INTERCONNECT PREVIEW
INTEL PROGRAMMABLE LOGIC SOFTWARE II (iPLS II)
01100100 10010010 JEDEC DESIGN FILE 01011100
LOGIC EQUATION FILE
ERROR MESSAGE FILE
RESOURCE UTILIZATION REPORT
LOGIC OPTIMIZING COMPILER
MACRO EXPANDER
ESPRESSO LOGIC MINIMIZER
RESOURCE FITTER
ELPD PARTS LIBRARY
ADVANCED DESIGN FILE (*ADF)
TTL or USER MACRO LIBRARIES
MACRO LIBRARIAN
MACRO DEVICE FILES
STATE MACHINE CONVERTER
SCHEMATIC FILE CONVERTER
STATE MACHINE FILE (*SMF)
SCHEMA II-PLD, iPLDDRAW, PCAD DASH, or FUTURENET NETLIST
TEXT EDITOR-NETLIST ENTRY
TEXT EDITOR-BOOLEAN EQUATION ENTRY
STATE MACHINE ENTRY
TTL or EPLD SCHEMATIC ENTRY

290134-2

Netlist Conversion

If schematic capture of state machine entry is used, the design must be converted into an ADF format. The optional SCHEMA II-PLD schematic capture package is a low-cost way to enter schematic designs. SCHEMA II-PLD supports EPLD primitives and TTL or user-defined macro symbols. It also outputs directly in ADF format. SCHEMA II-PLD contains the EPLD Manager, which provides a single user interface to both SCHEMA II-PLD and iPLS II software.

The P-CAD†† and Futurenet† systems may be used to capture EPLD symbols provided the EPLD libraries and ADF convertors are used. State machine entry may be performed via the iSTATE software and a standard text editor.

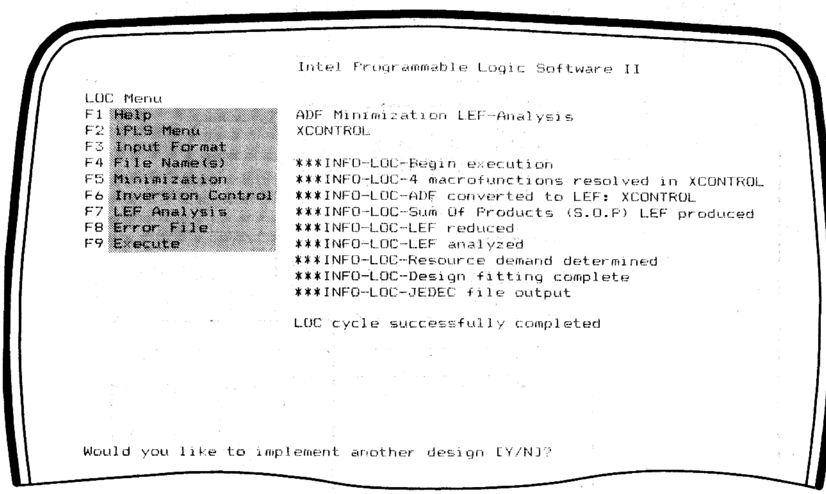
†Futurenet is a registered trademark of FutureNet Corporation.

††P-CAD is a registered trademark of P-CAD Corporation.

File Compilation

File compilation is performed by the LOGIC OPTIMIZING COMPILER. The LOC accepts an ADF and converts it into an industry standard JEDEC file which is used to program the device. As a part of this process, the LOC expands TTL macros into equivalent EPLD logic, minimizes the logic equations using the Espresso algorithm, and maps the network and logic equations into a cell map for the selected device. The final output of the LOC is a JEDEC Design File. The JEDEC Design File describes the input design for the designated EPLD in JEDEC standard format.

For designs using the 5AC312 or 5AC324 iPLS II R2.0 utilizes proprietary algorithms to efficiently use the device resources. The improved Fitter in R2.0 optimizes fitting for all devices.



290134-9

Logic Optimizing Compiler Main Menu

Device Programming

The programming hardware is controlled by the LOGIC PROGRAMMER SOFTWARE. LPS takes the JEDEC file produced by the LOC and programs it into the device. LPS can also read a programmed device or verify that a device has been programmed correctly.

The Intel Universal Programmer for the Personal Computer (iUP-PC) is a versatile programming solution in a PC-based system. Installed in an IBM PC/XT, PC/AT or compatible host, the iUP-PC emulates the performance of the standalone INTEL iUP-200A Universal Programmers. As such, it supports the iUP Generic Universal Programmer Interface (iUP-GUPI). With the appropriate socket adapters for the iUP-GUPI, the iUP-PC supports all Intel EPLDs. Future EPLDs will be supported by new GUPI adapters or adapter upgrades. Many other Intel devices—EPROMs, EEPROMs, and microcontrollers—are also supported by the GUPI. The iUP-PC is controlled by the LPS or the iPPS (Intel PROM Programmer Software). iPLDS II includes the iUP-PC, which contains the iPPS, PCPP programming card, interconnect cable, and the GUPI base. GUPI adapters are available separately.

iPLS II SOFTWARE

The Intel Programmable Logic Software II (iPLS II) has many options and enhancements for implementing a logic design. iPLS II accommodates a wide variety of design input methods. Schematics, state machines or Boolean equations may all be used provided the proper formats and convertors are implemented as needed. No matter what method is chosen, the Logic Optimizing Compiler will minimize and fit the design during compilation. Finally, iPLS II contains the Logic Programmer Software which controls the iUP-PC programming hardware for all Intel EPLDs.

I. Design Input

The entire spectrum of design input methods is available to the logic designer in iPLS II. Everything from TTL schematics to Boolean equations are accepted and processed by the LOC.

A. TTL SCHEMATIC ENTRY

SCHEMA II-PLD is an optional software package that allows EPLD design to be implemented with standard TTL functions. SCHEMA II-PLD contains a symbol library that includes common SSI/MSI TTL symbols. SCHEMA II-PLD also outputs directly in ADF format. The TTL symbols appear in the ADF in the form of macro calls. During compilation, iPLS II automatically expands these calls from its TTL macro library. Thus, with SCHEMA II-PLD, conversion to EPLD logic primitives is performed automatically in a manner completely transparent to the user.

Only parts supported by the SCHEMA II-PLD TTL symbol library and the iPLS II TTL macro definition library may be used for TTL schematic entry. In most cases, this won't be a limitation as the most common parts are included in both libraries. Parts not in the macro libraries may be created by the user and stored in proprietary user libraries. SCHEMA II-PLD also supports creating of user-defined macro symbols. The iPLS II Macro Librarian supports creation of iPLS II macro libraries.

B. SCHEMATIC ENTRY WITH INTEL SYMBOL LIBRARY

If the user prefers designing with EPLD logic primitives but still wants to use schematic entry, SCHEMA II-PLD, in addition to supporting TTL schematic capture, also supports design using EPLD primitive symbols. Users can enter their design and have both a schematic drawing and an ADF version of the design. The logic symbols are loaded from the Intel library and connected in the usual manner. For quicker use of EPLD primitives, a second library, EPLDMAC.LIB is available for use. Optional symbol libraries are also available for PC-CAPS* by P-CAD Corporation and DASH-2, -3, -4** by FutureNet (iSLIBPCAD, iSLIBFNET). The iSIMLIB optional library is available for simulating logic designs with P-CAD's PC-LOGS logic simulator.

*PC-CAPS and PC-LOGS are registered trademarks of P-CAD Corporation.

**DASH-2, -3, -4 are registered trademarks of FutureNet Corporation.

C. TEXT EDITOR ENTRY

Designers who are familiar with the logic primitives and the Advanced Design File format can directly enter ADFs with a standard text editor. The bulk of the design entry can be accomplished using Boolean Equations obtained from a Karnaugh map or truth table. Hence, the need for conversion to gates is eliminated. This method of entry is useful for sub-circuits that will be incorporated into larger designs.

D. STATE MACHINE ENTRY

In the past, state diagrams or flowcharts (ASM charts) were merely abstractions used to obtain the logic equations necessary to implement TTL designs. With the advent of the iPLS II state machine convertor (iSTATE), this is no longer the case. Using an IF THEN / ELSE format, the designer may enter the state machine description without having to extract the logic and convert the equations into TTL components. The state machine to Boolean logic conversion is handled by the state machine convertor, provided the input file adheres to the specified State Machine File (SMF) format.

Summary of Optional Entry Requirements:

TTL Schematic Capture

1. TTL Macro Library
2. EPLD Custom Macro Library
3. SCHEMA II-PLD

PC-CAPS

1. Intel Library used to design logic circuit
2. Component List Output
3. PCAD convertor used in LOC
   (Library and convertor contained in iSLIBPCAD)

DASH-2, -3, -4

1. Intel Library used to design logic circuit
2. Pin List Output
3. FutureNet convertor used in LOC
   (Library and convertor contained in iSLIBFNET)

State Machines

1. State Machine File (SMF) format used
2. Optional state machine convertor used in LOC
   (Convertor contained in iSTATE)

II. Logic File Compilation

Before programming the part, the designer must compile the input design file into a JEDEC standard file. This function is performed by the Logic Optimizing Compiler.

LOGIC OPTIMIZING COMPILER (LOC)

Once the input file is in Advanced Design File (ADF) format, the LOC will compile it into a device-specific JEDEC Design File. The first phase of this compilation is performed by the MACRO EXPANDER. The Macro Expander expands Intel or TTL macros into equivalent EPLD equations. The second phase is performed by the ESPRESSO MINIMIZER. The minimizer reduces all the logic equations to their simplest form using the ESPRESSO II-MV algorithm. The final phase of compilation is performed by the FITTER. The Fitter creates a cell map of the minimized equations according to the resources available within the specified device.

MACRO EXPANDER

The input design file is initially passed through the MACRO EXPANDER. The Macro Expander searches the file for any non-EPLD network elements. If found, the Expander then searches the User Libraries and TTL Library for the unidentified element. Once the element is located, the design file element is replaced by the equivalent EPLD primitive implementation found in the library. Having the Expander search the User Libraries allows the user to create his own macros. User macro files are created with a standard ASCII text editor and are stored in libraries by the iPLS II Macro Librarian.

ESPRESSO MINIMIZER

The minimization in the LOC is performed by the ESPRESSO II-MV MINIMIZER. Developed by the University of California at Berkeley, the ESPRESSO II-MV algorithm is regarded by many as being the best minimization method available. ESPRESSO II-MV uses DeMorgan's and other logic theorems to reduce the equations to the least number of product terms possible. Since product terms are the key variable in the EPLD architecture, the ESPRESSO II-MV Minimizer provides the simplest equations possible. As a result, the success rate for fitting large designs is dramatically increased.

FITTER

The FITTER examines the architecture of the specified device, then tries to map the minimized equations into the resources available. The Fitter automatically assigns pins unless pin assignments are

already specified in the design input file. The fitting sequence continues until a successful fit is accomplished or all possible implementations are exhausted. Release 2.0 of iPLS II includes a new, faster Fitter that supports PGA packages and the 5AC312, 5AC324, and 85C508. Also included in this new Fitter is the capability to allocate p-terms to adjacent macrocells for devices such as the 5AC312 and 5AC324 that support p-term allocation.

OUTPUT FILES

— JEDEC Design File
A properly designed circuit results in the desired file from the LOC—the JEDEC Design File. The JEDEC Design File is a device-tailored EPROM cell programming map expressed in JEDEC standard format.

— Resource Utilization Report
The Resource Utilization Report gives an in-depth view of what was used inside the EPLD. Items such as device pinout, macrocell usage, and feedback arrangements are all listed. Unused resources are also listed to aid the user in adding logic or merging EPLD designs.

— Logic Equation File
The LEF file lists the logic equations after they have passed through the minimizer. It is these equations that are actually implemented in the final design.

— Compiler Error File
If a logic circuit is incorrectly designed, messages are produced by the LOC denoting the errors. To assist the redesign, these errors are placed into the Compiler Error File for later reference.

FILE MERGING

Once a design is successfully implemented, the LOC can merge it with other designs by simultaneously running the two ADF's. In this manner, LSI circuits can be broken into manageable chunks that can be implemented and tested individually. After each portion is completed, the subcircuits can be merged into one ADF to implement the total design.

III. Device Programming

After the design has been successfully entered, minimized and fitted, the designer programs his part using the JEDEC file produced by the LOC. Programming is accomplished by running the Logic Programmer Software.

LOGIC PROGRAMMER SOFTWARE

To program a device with the LPS, the user enters the file name and device to be programmed. The LPS checks if the device is blank, programs the device, then verifies that the device was programmed correctly. As a part of the Intel EPLD Programming Algorithm, each programmed cell is checked. Adding the complete device check after programming gives double verification that the part has been successfully programmed.

It is also possible to read a pre-programmed device and program other devices with the program read. The JEDEC Editor in LPS provides a hierarchical view of the device from the pin level, to the macrocell level, to the product term level. At the product term level, individual EPROM cells may be set or reset to connect or disconnect the logic equation inputs.

If the user does not want an EPLD to be read, the Security bit may be set when running the LPS. The Security Bit prevents a device from being examined after it has been programmed. This function is useful for protecting confidential designs.

iUP-PC HARDWARE

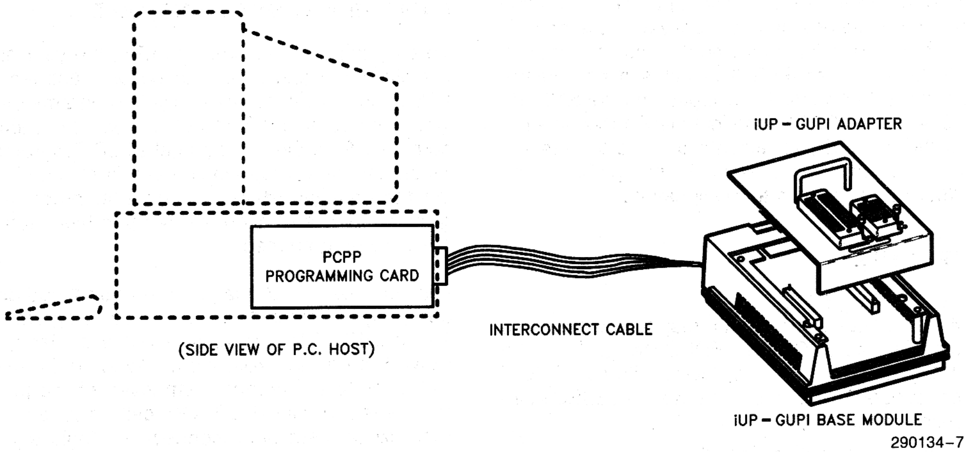
The Intel Universal Programmer for the Personal Computer consists of the PCPP programming card, 50-lead interconnect cable, GUPI base and GUPI adapter. Together they form a system for programming most PROM-type Intel devices directly from the PC host.

PCPP

The Personal Computer Personal Programmer (PCPP) is the programmer interface card that fits into the IBM AT/XT or true compatible. It is capable of driving both the iUP-GUPI base and the iUP-FAST27K personality module. The PCPP emulates the performance of the Intel iUP-200A. The LPS or iPPS (Intel PROM Programmer Software) controls the PCPP, causing the programming card to generate the control signals for the GUPI base.

GUPI BASE

The Generic Universal Programmer Interface (GUPI) is used for all programmable logic support. As all



The Intel Universal Programmer for the Personal Computer (iUP-PC)

signal generation to devices is done by the GUPI, the programming waveforms are extremely reliable. Using the GUPI also allows upgrading for future devices with the simple addition of a plug-in adaptor. Future Intel EPLDs will be supported by the GUPI system.

GUPI ADAPTERS

Table 1 details the GUPI adapters required for the logic devices. The adapters available for programming EPROM's, E2PROM's and microcontrollers can be found in the data sheet for the iUP-PC (Intel order number 290130). The adapters contain the device description data for a family of similar devices. New devices will be supported by new adapters or by upgrades to existing adapters.

SPECIFICATIONS

Host System

The iPLDS II software requires an IBM PC/XT, PC/AT or other true compatible computer capable of running MS-DOS* version 3.0 or later. The computer must have a 360KB double-sided, double-density diskdrive, a hard disk, and 512KB of RAM. Addi-

Table 1. Intel Programmable Logic Development System II Programming Support

Device	Number of Macrocells	iUP-GUPI Adapter	Package Type Supported
5C031, EP310	8	GUPI LOGIC-12	20 Pin DIP
5C032, EP320	8	GUPI LOGIC-12	20 Pin DIP
5C060, EP600	16	GUPI LOGIC-IID	24 Pin DIP
5C090, EP900	24	GUPI LOGIC-IID	40 Pin DIP
5C121, EP1200	28	GUPI LOGIC-12	40 Pin DIP
5C180, EP1800	48	GUPI LOGIC-18	68 Pin PLCC and JLCC
5C180PGA	48	GUPI LOGIC-18PGA	68 Pin PGA
5CBIC	(inPLU):8 (# of Ports):5	GUPI LOGIC-BIC	44 Pin PLCC
5AC312	12	GUPI LOGIC-IID	24 Pin DIP
5AC324	24	GUPI40D44J	40-Pin DIP
85C508	8	GUPI85EPLD28	28-Pin DIP and PLCC

(EPXXX Devices from Altera Corp.)

tional memory is recommended (640K) and is required for the optional schematic capture programs. A color monitor is recommended, as the color graphics available provide a better representation of the data than a monochrome display. The PCPP programming card requires one full-size card slot in the host computer. iPLSII (Intel Programmable Logic Software) can run on the IBM PS/2.

*MS-DOS is a trademark of Microsoft Corporation

Operating Environment

Electrical Characteristics

PCPP: Worst Case Power Consumption at IBM PC I/O Channel

Supply Voltage	Voltage Variance	Personality Module	Max. Current Drain
+5V	+5%, −4%	FAST27K	1.898 A
−12V	+10%, −9%	FAST27K	102.9 mA
+12V	+5%, −4%	GUPI	530 mA

Physical Characteristics

PCPP:

Length: 13.3 inches (33.9 cm)
Height: 3.9 inches (10.0 cm)

INTERCONNECT CABLE:

50 lead ribbon cable
Length: 3.0 feet (91.4 cm)
Width: 2.43 inches (5.5 cm)

GUPI:

Length: 7.0 inches (17.8 cm)
Width: 5.5 inches (1.4 cm)
Height: 1.6 inches (4.1 cm)

Environmental Characteristics

Operating Temperature: 10°C to 40°C
Operating Relative Humidity: 85% Maximum

Equipment Supplied

HARDWARE

— PCPP programming card
— Interconnect cable
— GUPI base
— (GUPI-LOGIC adaptors purchased separately)

SOFTWARE

— iPLS II
— iPPS
— PLDUTIL

DOCUMENTATION

— iPLS II User's Guide-V2.0 (order number 450196)
— PCPP User's Guide (order number 168161)

ORDERING INFORMATION

Order Code	Product Description
iPLDS II	Intel Programmable Logic Development System II: iPLS software, iUP-PC, iPLS II User's Guide
iPLS II	Intel Programmable Logic Software II: Logic Builder design entry, Logic Optimizing Compiler, Logic Programmer Software, iPLS II User's Guide
iUP-PC	Intel Universal Programmer for the Personal Computer: PCPP programming card, interconnect cable, iUP-GUPI base, Intel PROM Programming Software PCPP User's Guide
MLIB	iPLS II Macro Librarian: Macro Librarian Software and User's Guide Supplement for creating user-defined macro libraries.
iSTATE	Intel State Machine Software: Entry format documentation, state machine convertor for LOC
iSLIBFNET	Intel Symbol Library—FutureNet: EPLD symbol library for FutureNet DASH-2 schematic capture package, Futurenet Pinlist convertor for LOC
iSLIBPCAD	Intel Symbol Library—PCAD: EPLD symbol library for PCAD PC-CAPS schematic capture package, PCAD Component List convertor for LOC
iSIMLIB	Intel Simulation Library (PC-LOGS): EPLD simulation library for PC-LOGS simulator by PCAD
PLDUTIL	PLD Utilities: Functional Simulatory TTL Macro Library EPLD Custom Macro Library
iUP-GUPI	Intel Universal Programmer—Generic Universal Programmer Interface: Generic programmer base which holds GUPI adaptors
GUPI LOGIC-IID	GUPI Adaptor for the 5AC312, 5C060 and 5C090.
GUPI LOGIC-12	GUPI Adaptor for the 5C031, 5C032, 5C121 and future 20 DIP EPLDs
GUPI-LOGIC-18	GUPI Adaptor for the 5C180 and future 68 pin PLCC and JLCC EPLDs
GUPI LOGIC-18PGA	GUPI Adaptor for the 5C180 device in a 68 pin PGA package.
GUPI-LOGIC-BIC	GUPI Adaptor for the 5CBIC and follow-on products
GUPI40D44J	GUPI Adaptor for the 5AC324; includes 40-pin DIP and 44-pin JLCC sockets.
GUPI85EPLD28	GUPI Adaptor for the 85C508; includes 28-pin DIP and JLCC sockets.
ADAPT24TO28	Adapts 24 pin DIP socket to 28 pin PLCC socket; for use with GUPI LOGIC-09 and GUPI LOGIC-IID.
ADAPT40TO44	Adapts 40 pin DIP socket to 44 pin PLCC socket; for use with GUPI LOGIC-09 and GUPI LOGIC-IID.